リフト、スプラット、シュート：3Dへの暗黙の投影解除による任意カメラリグからの画像のエンコード

Jonah Philion　Sanja Fidler

NVIDIA　University of Toronto　Vector Institute

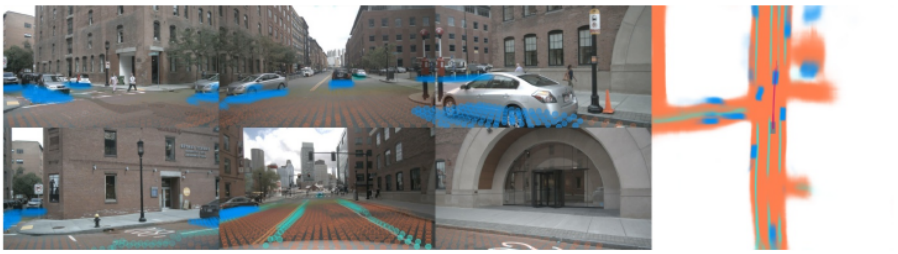


図1:マルチビューカメラデータ(左)が与えられたとき、鳥瞰(BEV)座標フレーム(右)で直接意味を推論するモデルを提案する。車両セグメンテーション(青)、走行可能領域(オレンジ)、車線セグメンテーション(緑)を示す。これらのBEV予測は、入力画像(左の点)に投影し直される。

概要。自律走行車の知覚の目標は、複数のセンサーから意味表現を抽出し、これらの表現を単一の「鳥瞰」座標フレームに融合して、運動計画による消費を行うことである。任意の数のカメラから画像データが与えられたシーンの鳥瞰表現を直接抽出する新しいエンドツーエンドアーキテクチャを提案する。我々のアプローチの核となる考え方は、各画像を個別に「持ち上げ」、各カメラの特徴量のフラストレーションにし、次にすべてのフラストレーションをラスタライズされた鳥瞰グリッドに「スプラット」することである。カメラリグ全体で学習することで、我々のモデルが画像を表現する方法だけでなく、キャリブレーションエラーに頑健でありながら、全てのカメラからの予測をシーンの単一のまとまった表現に融合する方法を学習できることを示す証拠を提供する。物体セグメンテーションや地図セグメンテーションなどの標準的な鳥瞰タスクにおいて、我々のモデルは全てのベースラインや先行研究を凌駕している。運動計画のための密な表現を学習するという目標を追求するために、我々のモデルによって推論された表現が、我々のネットワークによって出力された鳥瞰コストマップにテンプレート軌道を「撮影」することによって、解釈可能なエンドツーエンドの運動計画を可能にすることを示す。ライダーからのオラクル深度を使用するモデルに対して、我々のアプローチをベンチマークする。https://nv-tlabs.github.io/lift-splat-shoot.

1 Introduction

コンピュータビジョンアルゴリズムは、一般に、画像を入力とし、座標フレームに依存しない予測(分類[19,30,16,17]など)、または、物体検出、セマンティックセグメンテーション、パノプティックセグメンテーション[7,1,15,36]などの入力画像と同じ座標フレームにおける予測を出力する。
このパラダイムは、すぐに自動運転における知覚の設定と一致しない。自動運転では、複数のセンサーが入力として与えられ、それぞれが異なる座標フレームを持つ。知覚モデルは最終的に、図2に示すように、下流のプランナーが消費するために、新しい座標フレーム(エゴ・カーのフレーム)で予測を生成するタスクを課される。

単一画像パラダイムをマルチビュー設定に拡張するための、シンプルで実用的な戦略は数多くある。例えば、n台のカメラからの3D物体検出の問題では、すべての入力画像に個別に単一画像検出器を適用し、物体が検出されたカメラの固有と外在に従って、各検出を回転させ、自我フレームに変換することができる。このシングルビューパラダイムのマルチビュー設定への拡張は、3つの貴重な対称性をその中に焼き込んでいる。

1. 翻訳の等価性 - 画像内の画素座標がすべてシフトしている場合、出力は同じだけシフトする。完全畳み込み単一画像物体検出器は、おおよそこの性質を持ち、多視点拡張はこの性質をそこから継承している[11] [6]。
2. **Permutation invariance** – the final output does not depend on a specific ordering of the n cameras.
3. 自我フレーム等角等価性 - 画像を取り込んだカメラが自我の車に対して相対的にどこに位置していても、与えられた画像から同じ物体が検出される。この性質を述べる等価な方法は、エゴフレームの定義を回転/並進させることができ、出力はそれを使って回転/並進させることができるということである。

上記の単純なアプローチの欠点は、単一画像検出器からの後処理された検出を使用することで、センサー入力に戻る途中のエゴフレームで行われた予測との区別がつかないことである。その結果、モデルはデータ駆動型の方法で、カメラ間で情報を融合させるのが最良の方法であることを学習することができない。また、バックプロパゲーションは、下流のプランナーからのフィードバックを用いて知覚システムを自動改善するために使用することができないことを意味する。
我々は、上記で特定した3つの対称性を設計上保持しつつ、エンドツーエンドで微分可能な「Lift-Splat」と名付けたモデルを提案する。セクション3では、我々のモデルが、下流のタスクである運動計画に便利なように、文脈的特徴のフラストラム形状の点群を生成することによって、画像を3Dに「持ち上げる」方法について説明する。セクション3.3では、解釈可能なエンドツーエンドのモーションプランニングのために、提案軌道をこの基準平面に「撮影」する方法を提案する。セクション4では、フルカメラリグでリフトスプラットモデルを効率的に学習するための実装の詳細を示す。

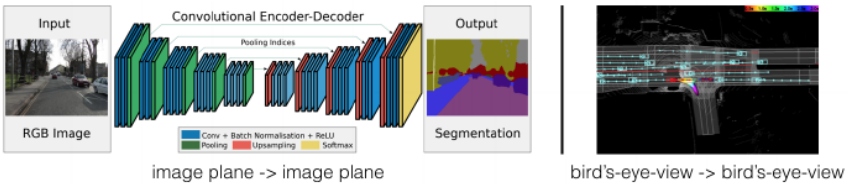


図2:(左、SegNet [1]より)従来、セマンティックセグメンテーションのようなコンピュータビジョンのタスクは、入力画像と同じ座標フレームで予測を行うものであった。(右、Neural Motion Planner [41]より) 対照的に、自動運転の計画は一般的に鳥瞰フレームで動作する。我々のモデルは、マルチビュー画像からエンドツーエンドで計画を立てるために、与えられた鳥瞰フレームで直接予測を行う。

第5節では、我々のモデルが可能な入力の分布から情報を融合するための効果的なメカニズムを学習するという経験的証拠を示す。

2 関連研究

複数のカメラからの画像データから凝集表現を学習するための我々のアプローチは、センサフュージョンと単眼物体検出における最近の研究に基づいている。Nutonomy[2]、Lyft[13]、Waymo[35]、Argo[3]の大規模なマルチモーダルデータセットにより、最近、カメラ入力のみを条件とする360*シーン全体の自車両に局所的な完全な表現学習が可能になった。Lift-Splatアーキテクチャでその可能性を探る。

2.1 単眼物体検出

単眼物体検出器は、画像平面から与えられた3次元参照フレームへの変換をどのようにモデル化するかによって定義される。標準的な手法は、成熟した2D物体検出器を画像平面に適用し、2Dボックスを3Dボックスに回帰する2つ目のネットワークを訓練することである[12,26,31,27]。nuScenesベンチマーク[31]における現在の最先端の3Dオブジェクト検出器は、標準的な2D検出器を訓練するアーキテクチャを使用し、誤ったバウンディングボックスによる誤差から誤った深さによる誤差を分離しようとする損失を使用して、深さも予測する。これらのアプローチは、3Dオブジェクト検出ベンチマークにおいて、画像平面での検出が、単眼の奥行き予測を覆い隠す基本的な曖昧さの雲を因数分解するため、優れた性能を達成する。
最近の経験的な成功を収めたアプローチは、単眼の奥行き予測を行うネットワークと、鳥瞰検出を行うネットワークを別々に訓練することである[39] [40]。これらのアプローチは「擬似ライダー」の名で呼ばれている。擬似ライダーが経験的に成功した直感的な理由は、擬似ライダーが、最終的に検出が評価され、画像平面に対してユークリッド距離がより意味のある座標フレームで動作する鳥瞰ネットワークの学習を可能にするからである。

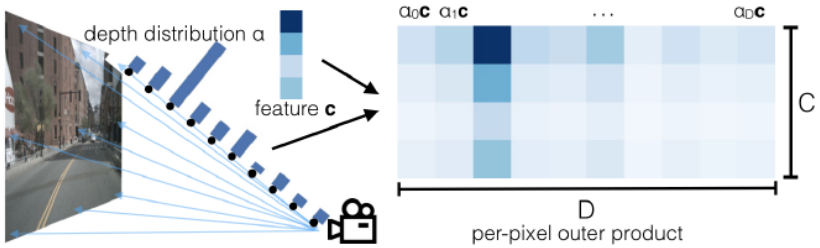


図3:モデルの「リフト」ステップを可視化したものである。各画素について、奥行き$\alpha \in 4^{D-1}$（左）とコンテキストベクトル$c \in^{C}_{R}$（左上）のカテゴリ分布を予測する。レイに沿った各点における特徴は、αとcの外積によって決定される（右）。

第3のカテゴリの単眼物体検出器は、3次元物体プリミティブを使用し、利用可能なすべてのカメラへの投影に基づいて特徴を取得する。Mono3D[4]は、利用可能な画像に投影することでスコアリングされる接地面上の3次元提案を生成することで、KITTI上で最先端の単眼物体検出を達成した。正射影特徴変換[29]は、Mono3Dをベースに、ボクセルの固定立方体を画像に投影して特徴を収集し、ボクセル内の特徴を条件として3Dで検出する2番目の「BEV」CNNを学習する。我々のモデルが扱うこれらのモデルの潜在的な性能ボトルネックは、ある画素がその画素のオブジェクトの深さに関係なく、すべてのボクセルに同じ特徴を寄与することである。

2.2 鳥瞰フレームにおける推論

鳥瞰フレームで直接推論を行うために、エクストリンシックとイントリンシックを使用するモデルは、最近大きな関心を集めている。MonoLayout[21]は、1枚の画像から鳥瞰推論を行い、敵対的損失を用いて、モデルがもっともらしい隠されたオブジェクトを塗りつぶすことを促す。同時進行で、Pyramid Occupancy Networks [28]は、画像表現を鳥瞰表現に変換する変換器アーキテクチャを提案している。FISHING Net [9]も、現在のタイムステップでオブジェクトをセグメント化し、将来予測を実行するマルチビューアーキテクチャを提案している。セクション5では、我々のモデルが先行研究を経験的に上回ることを示す。これらのアーキテクチャは、我々のアーキテクチャと同様に、機械学習グラフィックスコミュニティ[34,32,38,20]の「マルチプレーン」画像に似たデータ構造を使用している。

3 Method

本節では、任意のカメラリグで撮影された画像データから、シーンの鳥瞰表現を学習するためのアプローチを紹介する。1節で示した対称性を尊重するようにモデルを設計する。$\{I_{i,k} \in \mathbb{R}^{3\times H\times W}\}_n$が与えられ、次を求める。

BEV座標フレームy$\in \mathbb{R}^{C \times X \times Y}$におけるシーンのラスタライズ表現.外在行列と内在行列は、n台のカメラそれぞれについて、参照座標(x, y, z)から局所ピクセル座標(h, w, d)へのマッピングを定義する。学習時やテスト時に深度センサーにアクセスする必要はない。

3.1 リフト: 潜在的な奥行き分布

我々のモデルの第一段階は、カメラリグ内の各画像に対して単独で動作する。このステージの目的は、各画像を局所的な2次元座標系から、すべてのカメラで共有される3次元フレームに「持ち上げる」ことである。
単眼センサーフュージョンの課題は、参照フレーム座標に変換するために深度を必要とするが、各ピクセルに関連する「深度」は本質的に曖昧であることである。我々の提案する解決策は、各ピクセルに対して、すべての可能な深さの表現を生成することである。

X$\in \mathbb{R}^{3 \times H \times W}$を外在Eと内在Iを持つ画像とし、pを画像座標(h, w)を持つ画像中の画素とする。各画素に|D|点{(h, w, d) $\in \mathbb{R}^3$ | d∈D}を関連付ける。ここでDは離散的な深さの集合であり、例えば{d_0 + △, ・・・, d_0 + |D|*}で定義される。この変換には学習可能なパラメータがないことに注意。サイズ D - H - W の与えられた画像に対して、単純に大きな点群を作成する。この構造は、マルチビュー合成コミュニティ[38,32]がマルチプレーン画像と呼ぶものと等価であるが、我々の場合、各プレーンの特徴は(r, g, b, α)値ではなく抽象ベクトルである。
点群中の各点のコンテキストベクトルは、注意と離散深度推論の概念に一致するようにパラメータ化される。画素pにおいて、ネットワークは各画素について文脈c$\in \mathbb{R}^C$と深さ$\alpha \in 4^{|D|-1}$上の分布を予測する。次に、点p_dに関連する特徴$c_d \in \mathbb{R}^C$は、α_dでスケーリングされた画素pのコンテキストベクトルとして定義される:

$$\mathbf{c}_d = \alpha_d \mathbf{c}. \tag{1}$$

もし我々のネットワークがαのワンホットベクトルを予測するのであれば、点p_dのコンテキストは、擬似ライダー[39]のように、単一の深さd^*に対してのみ非ゼロになることに注意。ネットワークが深度上の一様分布を予測する場合、ネットワークはOFT [29]と同様に、深度に依存しないピクセルpに割り当てられた各点p_dに対して同じ表現を予測する。したがって、我々のネットワークは、理論的には、画像からのコンテキストを鳥瞰表現の特定の位置に配置するか、例えば奥行きが曖昧な場合、コンテキストを空間の光線全体に広げるかを選択することができる。
要約すると、理想的には、任意の空間位置で問い合わせ可能な各画像に対して、関数g_c : (x, y, z) $\in \mathbb{R}^3 \rightarrow$ c$\in \mathbb{R}^C$ を生成し、コンテキストベクトルを返したい。離散畳み込みを利用するために、空間を離散化する。カメラの場合、カメラに見える空間の体積はフラストレーションに相当する。図3にビジュアルを示す。

3.2 スプラット:柱状プーリング

pointpillars[18]アーキテクチャに従い、”lift ”ステップで出力される大きな点群を変換する。

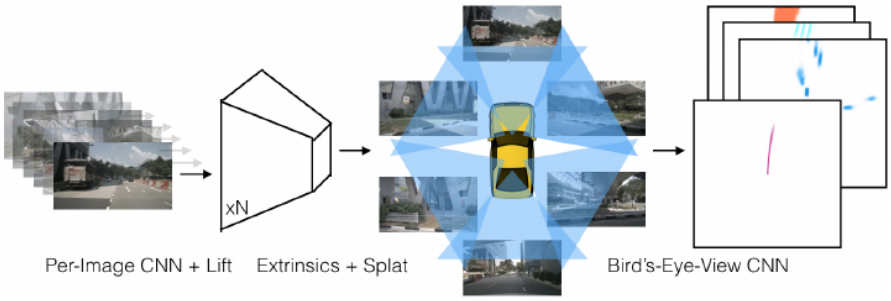


図4:Lift-Splat-Shootの概要 n枚の画像(左)とそれに対応する外部パラメータと内部パラメータを入力とするモデル。リフト」ステップでは、個々の画像に対してフラストラム状の点群が生成されます(中央左)。次に、extrinsicsとintrinsicsを使用して、各フラストレーションを鳥瞰図平面上にスプラットします(中央右)。最後に、鳥瞰CNNが鳥瞰表現を処理して、BEVの意味的セグメンテーションやプランニングを行う(右)。

「ピラー」は高さが無限のボクセルである。すべての点を最も近い柱に割り当て、和のプーリングを行い、標準的なCNNで処理できるC×H×Wテンソルを作成し、鳥瞰推論を行う。全体的なリフトスプラットアーキテクチャの概要を図4に示す。
OFT[29]がプーリングステップを高速化するために積分画像を使用するのと同様に、我々は和プーリングを高速化するためにアナラガス技術を適用する。生成される点群のサイズが大きい場合、効率はモデルを学習するために非常に重要である。各柱をパディングしてから和のプーリングを行う代わりに、パッキングを使い、和のプーリングに「和のトリック」を活用することで、パディングを避ける。この操作には解析的勾配があり、4.2節で説明したように、autogradを高速化するために効率的に計算することができる。

3.3 シュート: モーションプランニング

我々のLift-Splatモデルの重要な点は、カメラのみの入力から運動計画を行うためのエンドツーエンドのコストマップ学習を可能にすることである。テスト時に、推論されたコストマップを使用した計画は、異なる軌道を「撮影」し、そのコストをスコアリングし、最もコストの低い軌道に従って行動することで達成できる[25]。セクション5.6では、エンドツーエンドで解釈可能なモーションプランニングを可能にする我々のモデルの能力を調査し、ライダーベースのエンドツーエンドのニューラルモーションプランナーと性能を比較する。

我々は、「プランニング」を自車両のK個のテンプレート軌道上の分布を予測することとして捉えている。

$$\mathcal{T} = \{\tau_i\}_K = \{\{x_j, y_j, t_j\}_T\}_K$$

センサー観測値 p(τ |o)を条件とする。我々のアプローチは、最近提案されたニューラルモーションプランナー(NMP)[41]に触発されたもので、点群と高精細マップを条件として、提案された軌道をスコアリングするために使用できるコストボリュームを生成するアーキテクチャである。

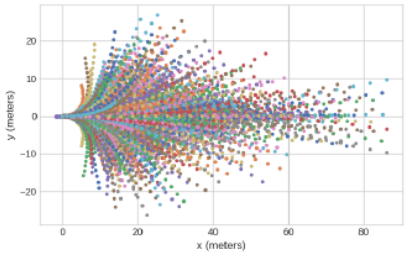


図5:トレーニング時とテスト時にコストマップに「撮影」する1Kの軌跡テンプレートを可視化したもの。学習中、各テンプレートの軌跡のコストは計算され、テンプレート上の1K次元ボルツマン分布として解釈される。テストでは、この分布のargmaxを選択し、選択したテンプレートに従って行動する。

NMPで提案されているハードマージン損失の代わりに、K個のテンプレート軌道の集合に対する分類としてプランニングを構成する。計画問題のコスト・ボリュームの性質を活用するために、K個のテンプレート軌道上の分布を以下の形に強制する。

$$p(\tau_i|o) = \frac{\exp\left(-\sum_{x_i,y_i\in\tau_i} c_o(x_i,y_i)\right)}{\sum_{\tau\in\mathcal{T}} \exp\left(-\sum_{x_i,y_i\in\tau} c_o(x_i,y_i)\right)} \tag{2}$$

ここで、c_o(x, y)は、位置x, yのオブザベーションoが与えられたときに予測されるコストマップにインデックスを付けることによって定義され、したがって、エキスパートの軌跡の対数確率を最適化することによって、データからエンドツーエンドで学習することができる。ラベルについては、真実の軌跡が与えられたとき、テンプレート軌跡Tに対するL2距離の最近傍を計算し、クロスエントロピー損失で学習する。この$p(\tau_i|o)$の定義により、NMP[41]のようにハードマージン損失を定義することなく、解釈可能な空間コスト関数を学習することができる。

実際には、多数のエキスパート軌道に対してK-Meansを実行することにより、テンプレート軌道の集合を決定する。実験に使用したコストマップへの「シュート」に使用したテンプレート軌道のセットを図5に可視化する。

4 Implementation

4.1 アーキテクチャの詳細

我々のモデルのニューラル・アーキテクチャはOFT [29]に似ている。OFTと同様に、我々のモデルは2つの大きなネットワークバックボーンを持つ。各画像から生成された点群を特徴付けるために、バックボーンの1つが各画像に対して個別に動作する。もう一つのバックボーンは、点群が基準フレーム内の柱にスプラットされた時点で、点群上で動作する。2つのネットワークは、セクション3で定義したように、リフトスプラット層によって結合され、図4で可視化される。

各画像に対して単独で動作するネットワークについては、ベースラインを含む全てのモデルについて、Imagenet [30]で事前学習したEfficientNet-B0 [37]のレイヤーを全ての実験で活用する。

EfficientNetsは、深さ、幅、解像度が比例してスケールアップされたリソース制限領域で、網羅的なアーキテクチャ探索によって発見されたネットワークアーキテクチャである。その結果、ResNet-18/34/50[8]と比較して、収束に多くの最適化ステップを必要とするという些細な不都合で、全てのモデルで高い性能を発揮できることがわかった。

鳥瞰ネットワークには、PointPillars [18]に似たResNetブロックの組み合わせを使用する。具体的には、カーネル7とストライド2の畳み込みに続いて、バッチノルム[10]とReLU[22]を行った後、ResNet-18の最初の3つのメタレイヤーを通過させ、異なる解像度x_1，x_2，x_3で3つの鳥瞰表現を得る。次に、x_3をスケールファクター4でアップサンプリングし、x_1と連結し、レスネットブロックを適用し、最後に2でアップサンプリングして、元の入力鳥瞰擬似画像の解像度に戻す。最終的なネットワークには14.3Mの学習可能なパラメータを数える。
我々のモデルの「解像度」を決定するハイパーパラメータがいくつかある。まず、入力画像のサイズ H × W が存在する。以下の全ての実験において、入力画像を128×352のサイズにリサイズして切り出し、それに応じて外在と内在を調整する。ネットワークのもう一つの重要なハイパーパラメータは、鳥瞰グリッドX × Yの解像度の大きさである。実験では、x、yともに-50mから50mまでのビンを設定し、セルの大きさは0.5m×0.5mとした。したがって、結果として得られるグリッドは200×200となる。最後に、ネットワークによって予測される深さの解像度を決定するDの選択がある。Dを1.0m間隔で4.0mから45.0mに制限する。これらのハイパーパラメータとアーキテクチャ設計の選択により、モデルのフォワードパスはTitan V GPU上で35hzで実行される。

4.2 フラストムプーリング累積和トリック

センサーリグ全体からのデータから学習するためには、学習効率が重要である。最大プーリングとは対照的に、セクション3では、我々の「累積和」トリックがパディングによる過剰なメモリ使用から私たちを節約するため、ピラー間で合計プーリングを選択する。累積和のトリック」とは、ビンIDに従ってすべての点をソートし、すべての特徴に対して累積和を実行し、ビンセクションの境界で累積和の値を引くことによって、和のプーリングを実行できるという観察である。3つのステップをバックプロップするためにautogradに依存する代わりに、モジュール全体の解析的勾配を導き出すことができ、学習を2倍高速化することができる。この層は、n枚の画像から生成されるフラストレーションを、カメラの数nに依存しない固定次元のC×H×Wテンソルに変換することができるため、「フラストレーションプーリング」と呼ぶことにする。コードはプロジェクトページで見ることができる。

5 実験と結果

nuScenes[2]とLyft Level 5[13]のデータセットを用いて、我々のアプローチを評価する。 nuScenesは、点群データと1kシーンの画像データからなる大規模なデータセットであり、それぞれ20秒の長さである。

両データセットのカメラリグは、前方、前方左、前方右、後方左、後方右、後方方向をほぼ指す6台のカメラで構成されている。すべてのデータセットにおいて、カメラの視野はわずかに重なっている。カメラの外在的パラメータと内在的パラメータは、両データセットを通してシフトしている。我々のモデルはカメラのキャリブレーションを条件としているため、これらのシフトを扱うことができる。
2つのオブジェクトベースのセグメンテーションタスクと2つのマップベースのタスクを定義する。物体分割タスクでは、3次元バウンディングボックスを鳥瞰平面に投影することで、グランドトゥルースの鳥瞰ターゲットを得る。nuScenes上の車セグメンテーションは、class vehicle.carのすべてのバウンディングボックスを指し、nuScenes上の車セグメンテーションは、メタカテゴリ車両のすべてのバウンディングボックスを指す。Lyft上の車セグメンテーションはクラスcarのすべてのバウンディングボックスを指し、nuScenes上の車両セグメンテーションはクラス∈{ car, truck, other_vehicle, bus, bike }のすべてのバウンディングボックスを指す。マッピングには、提供された6自由度ローカライゼーションとラスタライズを用いて、nuScenesマップからエゴフレームへの変換マップレイヤーを使用する。

すべてのオブジェクトセグメンテーションタスクにおいて、正の重み1.0を持つバイナリクロスエントロピーで学習する。車線セグメンテーションでは正の重みを5.0に設定し、道路セグメンテーションでは正の重みを1.0に設定する[24]。すべての場合において、Adam [14]を用いて、学習率1e − 3、重み減衰1e − 7で300kステップの学習を行う。PyTorchフレームワーク[23]を使用する。
Lyftデータセットには、正規のtrain/val分割は含まれていない。Lyftシーンの48シーンを検証用に分離し、nuScenesとほぼ同じサイズの検証セット(Lyftは6048サンプル、nuScenesは6019サンプル)を得る。

5.1 ベースラインの説明

バニラCNNとは異なり、我々のモデルは初期化時に3次元構造を備えている。この構造が性能向上のために重要であることは、標準的なモジュールで構成されたCNNと比較することで示される。我々はMonoLayout[21]に似たアーキテクチャに従っており、画像のみから鳥瞰ラベルを出力するようにCNNを訓練するが、アーキテクチャの設計に誘導バイアスを活用せず、単一カメラのみで訓練する。このアーキテクチャは、すべての画像にわたって独立に特徴を抽出するEfficientNet-B0バックボーンを持つ。表現を連結し、バイリニア補間を行い、モデルが出力するように$\frac{X}{R} \times \frac{Y}{R}$テンソルにアップサンプリングする。ネットワークは、我々のモデルとほぼ同じ数のパラメータを持つように設計する。このベースラインの性能の低さは、マルチビュー設定において、Sec 1の対称性3をモデルに明示的に焼き付けることがいかに重要であるかを示している。
我々のモデルが有用な暗黙の深さを予測していることを示すために、事前学習されたCNNの重みをOFT[29]と同様に凍結した我々のモデルと比較する。表1と表2に示すように、全てのタスクにおいて、これらのベースラインを上回った。また、同じセグメンテーションタスクでベンチマークを行う並行作業[9] [28]も上回った。その結果、アーキテクチャは、効果的な深度分布と、下流タスクのための効果的な文脈表現の両方を学習する。

	nuScenes		Lyft	
	Car	Vehicles	Car	Vehicles
CNN	22.78	24.25	30.71	31.91
Frozen Encoder	25.51	26.83	35.28	32.42
OFT	29.72	30.05	39.48	40.43
Lift-Splat (Us)	**32.06**	**32.07**	**43.09**	**44.64**
PON* [28]	24.7	-	-	-
FISHING* [9]	-	30.0	-	56.0

表1:セグメント BEVフレームにおけるIOU

	Drivable Area	Lane Boundary
CNN	68.96	16.51
Frozen Encoder	61.62	16.95
OFT	71.69	18.07
Lift-Splat (Us)	**72.94**	**19.96**
PON* [28]	60.4	-

表2:BEVフレームにおけるマップIOU

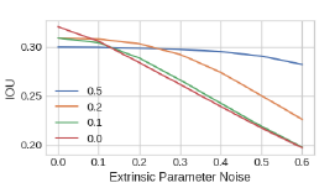
5.2 Segmentation

我々は、我々のLift-Splatモデルが、鳥瞰フレームにおける監視を与えられた意味的な3D表現を学習できることを実証する。物体分割タスクの結果を表1に、地図分割タスクの結果を表2に示す。すべてのベンチマークにおいて、我々はベースラインを上回った。我々は、特にオブジェクトのセグメンテーションにおいて、暗黙のうちに3Dに投影しないことによる性能向上の程度は相当なものであると考えている。また、2つの同時並行研究[9] [28]のIOUスコアも報告しているが、これらの論文はいずれも鳥瞰グリッドの定義が異なり、Lyftデータセットの検証分割も異なるため、真の比較はまだ不可能である。

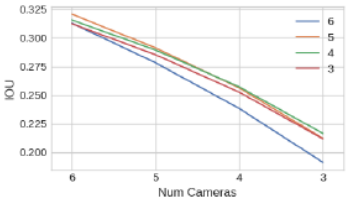
5.3 ロバスト性

鳥瞰CNNはデータからカメラ間で情報を融合する方法を学習するため、外在論者が偏ったり、カメラが死亡したりするような自動運転で発生する単純なノイズモデルに対してロバストであるようにモデルを訓練することができる。図6では、学習時にカメラを落とすことで、テスト時に落とすカメラをより良く扱うことができることを検証している。実際、6台のカメラがすべて存在する場合に最も性能が良いモデルは、学習中に各サンプルから1台のカメラをランダムに取り除いて学習させたモデルである。我々は、センサーのドロップアウトが、ドロップアウトの他のバリエーション[33] [5]と同様に、異なるカメラ上の画像間の相関をモデルに学習させると推論する。図6の左側に、ノイズの多いエクストリンシックでモデルをトレーニングすることで、テスト時のパフォーマンスが向上することを示す。テスト時のノイズ量が少ない場合、BEV CNNはスプラットの位置をより信頼できるため、エクストリンシックにノイズを含まないモデルが最も良い性能を発揮する。大量の外部ノイズに対して、我々のモデルは良好な性能を維持する。

図7では、nuScenes上での車のセグメンテーションの性能に対する各カメラの「重要度」を測定している。nuScenesでカメラを失うことは、車に近い領域の特定の領域がセンサー測定値を持たないことを意味し、その結果、性能は完全なセンサーリグでの性能に厳密に上界されることに注意。カメラの欠落によるネットワークの塗りつぶしの定性的な例を図 8 に示す。このようにして、各カメラの重要度を測定し、センサーの冗長性が安全性にとってより重要であることを示唆している。



(a) テスト時間外来ノイズ



(b) テスト時間カメラのドロップアウト

図6:一般的なセンサーエラーの原因に強いように、我々のネットワークを訓練することが可能であることを示す。左側は、エクストリンシック(青)のノイズを多量に用いて学習することで、テスト時にネットワークがエクストリンシックノイズに対してより頑健になることを示している。右図は、学習時に各バッチからカメラをランダムに落とすと(赤)、テスト時にセンサーのドロップアウトに対するロバスト性が向上することを示している。

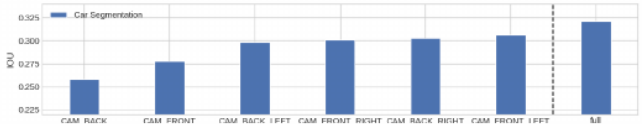


図7:各カメラが欠落している場合の、交差点-交差点間の車分分割を測定する。nuScenesカメラリグの後方カメラは視野が広いので、このカメラを失うと、フルカメラリグ(右側の「フル」と表示)での性能に対する性能の相対的な低下が最も大きくなることが直感的に理解できる。

	IOU
4	26.53
$4+1_{fl}$	27.35
$4+1_{bl}$	27.27
$4+1_{bl}+1_{fl}$	**27.94**

表3: nuScenesデータセットの6台のカメラのうち4台の画像のみで学習。次に、新しいカメラ(1_{bl}は「左後方」のカメラ、1_{fl}は「左前方」のカメラに対応)で評価したところ、学習時に未見のセンサーを追加するほど、モデルの性能が厳密に向上することがわかった。

5.4 ゼロショットカメラリグ転送

次に、Lift-Splatの汎化能力を探る。最初の実験では、nuScenesカメラリグのカメラのサブセットからの画像のみで学習し、テスト時に残りの2つのカメラからの画像にアクセスできる場合のモデルの性能を測定する。表3では、再トレーニングなしでテスト時にカメラを追加利用できる場合、車種分割のための我々のモデルの性能が向上することを示している。

上記の実験を少し進め、nuScenesのデータのみで学習させた場合、我々のモデルがLyftカメラリグにどの程度汎化されるかを探る。ベースラインの汎化に対するベンチマークを表 4 に示す。

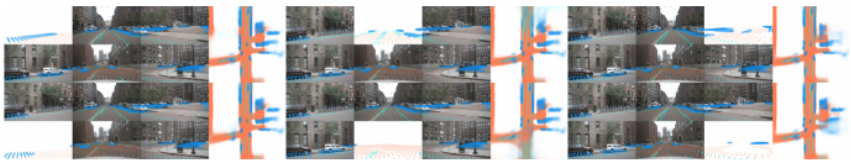


図8:1つのタイムスタンプについて、各カメラを削除し、カメラの損失がネットワークの予測にどのように影響するかを可視化する。欠損カメラでカバーされる領域は、どのケースでもファジーになる。前面カメラが取り除かれると(上中)、ネットワークはエゴの前の車線と走行可能領域を外挿し、右上のカメラでコーナーしか見えない車体の体を外挿する。

表4: nuScenesでモデルを学習し、Lyftで評価する。LyftカメラはnuScenesカメラとは全く異なるが、モデルはベースラインよりもはるかに優れた汎化に成功している。我々のモデルは、表1と表2の標準的なベンチマークからギャップを広げていることに注意。

	Lyft Car	Lyft Vehicle
CNN	7.00	8.06
Frozen Encoder	15.08	15.82
OFT	16.25	16.27
Lift-Splat (Us)	**21.35**	**22.59**

5.5 オラクル深度に対するベンチマーク

我々は、LIDAR点群からのグランドトゥルース深度を使用するpointpillars [18]アーキテクチャに対して、我々のモデルをベンチマークする。表5に示すように、全てのタスクにおいて、我々のアーキテクチャは、LIDARのシングルスキャンで学習した点ピラーよりもわずかに性能が悪い。しかし、少なくとも走行可能領域のセグメンテーションでは、LIDARの性能に近づくことに注意する。一般的に、すべてのレーンがライダースキャンで見えるわけではない。今後、より広範な環境におけるパフォーマンスを測定したい。

我々のモデルがLIDARとどのように異なるかを知るために、自車両までの距離と天候という2つの制御変数によって、車のセグメンテーションの性能がどのように変化するかをプロットする。nuScenesデータセットの各シーントークンに付随する記述文字列から、シーンの天候を決定する。結果を図10に示す。予想通り、夜間に発生するシーンでは、我々のモデルの性能はポイントピラーよりもはるかに悪いことがわかる。また、両モデルとも深さが増すにつれて、ほぼ直線的な性能低下を経験することがわかった。

5.6 モーションプランニング

最後に、Lift-Splatが出力する表現をコスト関数として学習させることで、我々のモデルがプランニングを行う能力を評価する。生成する軌道は、0.25秒間隔で5秒である。テンプレートを取得するために、K = 1000のK-MeansをnuScenesの学習セット内の全てのエゴの軌跡に適合させる。

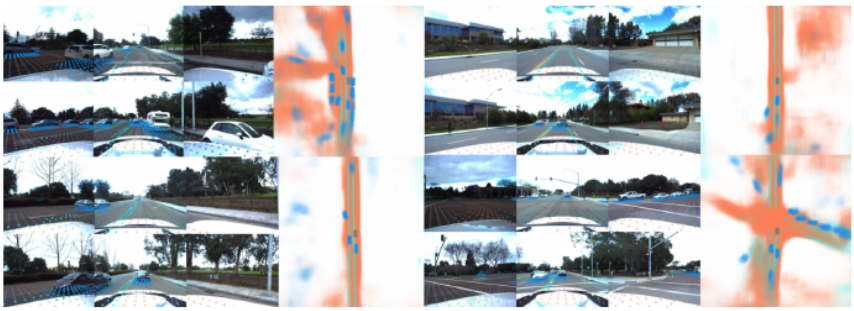


図9:テスト時に全く新しいカメラリグが与えられた場合、我々のモデルがどのように機能するかを定性的に示す。道路セグメンテーションはオレンジ色、車線セグメンテーションは緑色、車両セグメンテーションは青色で示されている。

			nuScenes		Lyft	
	Drivable Area	Lane Boundary	Car	Vehicle	Car	Vehicle
Oracle Depth (1 scan)	74.91	25.12	40.26	44.48	74.96	76.16
Oracle Depth (> 1 scan)	76.96	26.80	45.36	49.51	75.42	76.49
Lift-Splat (Us)	70.81	19.58	32.06	32.07	43.09	44.64

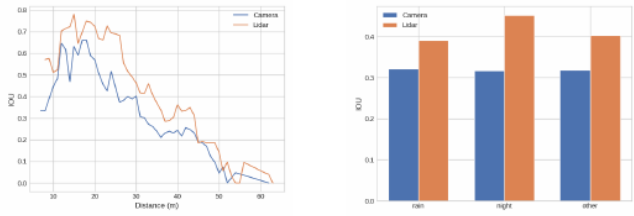
表5:ライダーからのオラクル深度を使用するモデルと比較すると、まだ改善の余地がある。ライダーを超えるために必要な奥行き推定値を取得するためには、カメラリグからのビデオ推論が必要と思われる。

テスト時に、L2ノルムの下で、ネットワークがグランドトゥルースの軌跡に最も近いテンプレートをどれだけ予測できるかを測定する。この実験のグランドトゥルースターゲットは、グランドトゥルースの3Dバウンディングボックスよりも取得コストが桁違いに低いため、このタスクは自動運転にとって重要な実験である。カメラベースのアプローチとライダーベースのアプローチの性能比較のベンチマークにとっても、このタスクは重要である。なぜなら、カメラのみからの3Dオブジェクト検出の上限は、ライダーのみによって確実に上限が設定されるが、カメラのみを使用した最適プランナは、原理的にはライダーのみから学習した最適プランナの性能を上限とすべきであるからである。

計画実験の定性的結果を図11に示す。点柱に対してベンチマークを行った実証結果を表6に示す。出力軌跡は、道路境界線をたどったり、横断歩道やブレーキ車両の後方で停止したりするような望ましい挙動を示している。

6 Conclusion

本研究では、任意のカメラリグから鳥瞰表現を推論するために設計されたアーキテクチャを紹介する。我々のモデルは、モデルの能力を調査するために設計された一連のベンチマークセグメンテーションタスクにおいて、ベースラインを上回った。



(a) 距離に対するIOU (b) IOU対天候

図10:我々のモデルの性能が深度と天候によってどのように変化するかを比較する。予想通り、我々のモデルは夜間にポイントピラーに比べて性能が低下する。

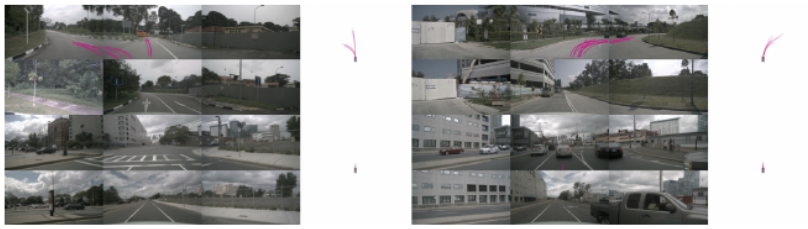


図11:1k個のテンプレートのうち、上位10位までの軌跡を表示。ビデオシーケンスはプロジェクトページで提供されている。我々のモデルは、1つのタイムスタンプからの観測値から二峰性の分布と曲線を予測する。我々のモデルは車の速度にアクセスできないので、横断歩道やブレーキライト付近の低速軌道を予測することは説得力がある。

	Top 5	Top 10	Top 20
Lidar (1 scan)	19.27	28.88	41.93
Lidar (10 scans)	24.99	35.39	49.84
Lift-Splat (Us)	15.52	19.94	27.99

表6:プランニングは1Kのテンプレート軌道の集合の分類として組み立てられているため、トップ5、トップ10、トップ20の精度を測定する。我々のモデルは、汎化においてライダーベースのアプローチにまだ遅れをとっていることが分かる。我々のモデルが出力する軌道の定性的な例を図11に示す。

は、訓練時やテスト時にグランドトゥルースの深度データにアクセスすることなく、鳥瞰フレームでセマンティクスを表現することができる。本論文では、キャリブレーションノイズの単純なモデルに対して、ネットワークを頑健にするモデルの学習方法を示す。最後に、このモデルにより、軌跡撮影のパラダイムに従ったエンドツーエンドのモーションプランニングが可能になることを示す。点群からのグランドトゥルース深度データのみを使用する類似のネットワークの性能を満たし、場合によってはそれを上回るためには、今後の研究では、本研究で検討するように、単一の時間ステップではなく、画像の複数の時間ステップを条件とする必要がある。

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR **abs/1511.00561** (2015), http://arxiv.org/abs/1511.00561
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. CoRR **abs/1903.11027** (2019), http://arxiv.org/abs/1903.11027
3. Chang, M.F., Ramanan, D., Hays, J., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., et al.: Argoverse: 3d tracking and forecasting with rich maps. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2019)
4. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2147–2156 (06 2016)
5. Ghiasi, G., Lin, T., Le, Q.V.: Dropblock: A regularization method for convolutional networks. CoRR **abs/1810.12890** (2018), http://arxiv.org/abs/1810.12890
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR **abs/1703.06870** (2017), http://arxiv.org/abs/1703.06870
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
9. Hendy, N., Sloan, C., Tian, F., Duan, P., Charchut, N., Xie, Y., Wang, C., Philbin, J.: Fishing net: Future inference of semantic heatmaps in grids (2020)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR **abs/1502.03167** (2015), http://arxiv.org/abs/1502.03167
11. Kayhan, O.S., Gemert, J.C.v.: On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
12. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. CoRR **abs/1711.10006** (2017)
13. Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P., Omari, S., Shah, S., Kulkarni, A., Kazakova, A., Tao, C., Platinsky, L., Jiang, W., Shet, V.: Lyft level 5 av dataset 2019. urlhttps://level5.lyft.com/dataset/ (2019)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
15. Kirillov, A., He, K., Girshick, R.B., Rother, C., Dollár, P.: Panoptic segmentation. CoRR **abs/1801.00868** (2018), http://arxiv.org/abs/1801.00868
16. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

18. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. CoRR **abs/1812.05784** (2018)
19. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. pp. 2278–2324 (1998)
20. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes. ACM Transactions on Graphics **38**(4), 114 (Jul 2019). https://doi.org/10.1145/3306346.3323020, `http://dx.doi.org/10.1145/3306346.3323020`
21. Mani, K., Daga, S., Garg, S., Shankar, N.S., Jatavallabhula, K.M., Krishna, K.M.: Monolayout: Amodal scene layout from a single image. ArXiv **abs/2002.08394** (2020)
22. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML (2010)
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
24. Philion, J.: Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
25. Philion, J., Kar, A., Fidler, S.: Learning to evaluate perception models using planner-centric metrics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
26. Poirson, P., Ammirato, P., Fu, C., Liu, W., Kosecka, J., Berg, A.C.: Fast single shot detection and pose estimation. CoRR **abs/1609.05590** (2016)
27. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 8851–8858 (07 2019). https://doi.org/10.1609/aaai.v33i01.33018851
28. Roddick, T., Cipolla, R.: Predicting semantic map representations from images using pyramid occupancy networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
29. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. CoRR **abs/1811.08188** (2018)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2014)
31. Simonelli, A., Bulò, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. CoRR **abs/1905.12365** (2019)
32. Srinivasan, P.P., Mildenhall, B., Tancik, M., Barron, J.T., Tucker, R., Snavely, N.: Lighthouse: Predicting lighting volumes for spatially-coherent illumination (2020)
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
34. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. CoRR **abs/1802.08275** (2018), `http://arxiv.org/abs/1802.08275`
35. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J.,

Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset (2019)
36. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
37. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. CoRR **abs/1905.11946** (2019), `http://arxiv.org/abs/1905.11946`
38. Tucker, R., Snavely, N.: Single-view view synthesis with multiplane images (2020)
39. Wang, Y., Chao, W., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. CoRR **abs/1812.07179** (2018)
40. You, Y., Wang, Y., Chao, W., Garg, D., Pleiss, G., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. CoRR **abs/1906.06310** (2019)
41. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8652–8661 (2019)